SYMBOL Research Computer
(Spring Joint Computer Conference Session #34)

Fairchild Camera & Instrument Corp.
464 Ellis Street
Mountain View, California 94040
Jack Leone: (415) 962-2391

Release No. 48-71


MAIN FRAME POWER CONTROL
SYSTEM MODE
SYSTEM CONTROL
CHANNEL STATUS
BASIC CLOCK CONTROL
MAINTENANCE UNIT CONTROL

SJ 71

FAIRCHILD

CAMERA AND INSTRUMENT CORPORATION

CORPORATE HEADQUARTERS
464 Ellis Street
Mountain View, California 94040
Jack Leone: (415) 962-2391

FOR IMMEDIATE RELEASE

FACT SHEET

Subject:	SYMBOL Research Computer
Developed by:	Fairchild Camera & Instrument Corp.
Project Leaders:	Rex Rice, manager, Digital Systems Research Department, Fairchild Camera & Instrument Corp.
	William R. Smith, project engineer - SYMBOL, Fairchild Camera & Instrument Corp.
SJCC Presentation:	A coordinated set of four papers will be presented at the Spring Joint Computer Conference Session #34, to describe many of the SYMBOL system's key elements.

SUMMARY DESCRIPTION:

The SYMBOL system was developed with Fairchild resources in order to provide a vehicle for Advanced Digital Systems Research. The computer's architecture features a hardware-implemented high level language as well as hardware-implemented dynamic data management. These represent a major departure from conventional computer systems in architectural philosophy.

-more-

SYMBOL is primarily directed to the problems of processing string, alphanumeric, dynamically varying data in large alphanumeric data bases.

The completed SYMBOL system was acquired by Iowa State University under a National Science Foundation grant for evaluation and continuing research. Professors R. M. Stewart, Jr. and R. J. Zingg are principal investigators for the Iowa State research project.

SYMBOL implements the following features directly in hardware:

Dynamic Memory Allocation
Dynamic Memory Reclamation
Dynamically Variable Field Lengths
Dynamically Variable Structures
Automatic Virtual Memory Management
Automatic Data Type Conversion
Automatic Time-sharing Supervision
Direct Symbolic Addressing
Precision-Controlled Arithmetic Processing
Direct Hardware Compilation
Alphanumeric Field Manipulation
Direct Text Editing

A very high level, general purpose procedural language is implemented in SYMBOL. Its development was based on a study of most modern computer languages including ALGOL, FORTRAN, PL/1, LISP and EULER. Special emphasis was placed on making the language artifact free.

-more-

SYMBOL achieves parallelism and execution efficiency by using time-sharing with multiprogramming and multiprocessing done through seven simultaneously operating autonomous processing units sharing a common virtual memory. One of the processing units is the direct implementation of a large portion of a time-sharing, multiprogramming, multiprocessing operating system.

The SYMBOL system accepts either fixed or floating point decimal numbers with positive or negative mantissas varying from one to 99 digits, with or without decimal point. Exponents, if any, may contain a plus or minus sign and may have up to two decimal digits. The operands for arithmetic may be fixed, floating, or mixed. A dynamically alterable limit register is provided to truncate computed results to a desired number of significant digits. The machine automatically identifies results that have been truncated.

Techniques have been developed in SYMBOL which allow complete and dynamic variability in the length of a string of characters used as a name, a word or a field. Each user is under no software/hardware constraint and may use any word or field size he wishes. It is not necessary to predetermine field size by declarations. This variability is provided in both the source program and in the data base.

SYMBOL includes direct hardware implementation of two operators to reconfigure string fields into desired formats. The FORMAT operator is used to manipulate numeric operands by applying a "pictorial string" mask against the operand field. The MASK operator provides a similar manipulation capability for alphanumeric string fields.

Complete dynamic variability of field size, vector size and structural configuration is provided at execute time. This is directly implemented in hardware to relieve the programmer from any necessity of declaring data base sizes and attributes.

SYMBOL demonstrates that even with complete dynamic variability in data fields, data structures and depth of structural nesting, it is possible to achieve execution speeds which are competitive with more conventional machines. In many areas such as compilation, output data formatting and paging, extremely high performance rates can be demonstrated.

-O-

Release No. 48-71

SJ 71

FAIRCHILD
CAMERA AND INSTRUMENT
CORPORATION

CORPORATE HEADQUARTERS
464 Ellis Street
Mountain View, California 94040
Jack Leone: (415) 962-2391

FOR IMMEDIATE RELEASE

FACT SHEET

Subject:	SYMBOL Research Computer
Developed by:	Fairchild Camera & Instrument Corp.
Project Leaders:	Rex Rice, manager, Digital Systems Research Department, Fairchild Camera & Instrument Corp. William R. Smith, project engineer - SYMBOL, Fairchild Camera & Instrument Corp.
SJCC Presentation:	A coordinated set of four papers will be presented at the Spring Joint Computer Conference Session #34, to describe many of the SYMBOL system's key elements.

SUMMARY DESCRIPTION:

The SYMBOL system was developed with Fairchild resources in order to provide a vehicle for Advanced Digital Systems Research. The computer's architecture features a hardware-implemented high level language as well as hardware-implemented dynamic data management. These represent a major departure from conventional computer systems in architectural philosophy.

-more-

SYMBOL is primarily directed to the problems of processing string, alphanumeric, dynamically varying data in large alphanumeric data bases.

The completed SYMBOL system was acquired by Iowa State University under a National Science Foundation grant for evaluation and continuing research. Professors R. M. Stewart, Jr. and R. J. Zingg are principal investigators for the Iowa State research project.

SYMBOL implements the following features directly in hardware:

Dynamic Memory Allocation
Dynamic Memory Reclamation
Dynamically Variable Field Lengths
Dynamically Variable Structures
Automatic Virtual Memory Management
Automatic Data Type Conversion
Automatic Time-sharing Supervision
Direct Symbolic Addressing
Precision-Controlled Arithmetic Processing
Direct Hardware Compilation
Alphanumeric Field Manipulation
Direct Text Editing

A very high level, general purpose procedural language is implemented in SYMBOL. Its development was based on a study of most modern computer languages including ALGOL, FORTRAN, PL/1, LISP and EULER. Special emphasis was placed on making the language artifact free.

-more-

SYMBOL achieves parallelism and execution efficiency by using time-sharing with multiprogramming and multiprocessing done through seven simultaneously operating autonomous processing units sharing a common virtual memory. One of the processing units is the direct implementation of a large portion of a time-sharing, multiprogramming, multiprocessing operating system.

The SYMBOL system accepts either fixed or floating point decimal numbers with positive or negative mantissas varying from one to 99 digits, with or without decimal point. Exponents, if any, may contain a plus or minus sign and may have up to two decimal digits. The operands for arithmetic may be fixed, floating, or mixed. A dynamically alterable limit register is provided to truncate computed results to a desired number of significant digits. The machine automatically identifies results that have been truncated.

Techniques have been developed in SYMBOL which allow complete and dynamic variability in the length of a string of characters used as a name, a word or a field. Each user is under no software/hardware constraint and may use any word or field size he wishes. It is not necessary to predetermine field size by declarations. This variability is provided in both the source program and in the data base.

SYMBOL includes direct hardware implementation of two operators to reconfigure string fields into desired formats. The FORMAT operator is used to manipulate numeric operands by applying a "pictorial string" mask against the operand field. The MASK operator provides a similar manipulation capability for alphanumeric string fields.

-more-

Complete dynamic variability of field size, vector size and structural configuration is provided at execute time. This is directly implemented in hardware to relieve the programmer from any necessity of declaring data base sizes and attributes.

SYMBOL demonstrates that even with complete dynamic variability in data fields, data structures and depth of structural nesting, it is possible to achieve execution speeds which are competitive with more conventional machines. In many areas such as compilation, output data formatting and paging, extremely high performance rates can be demonstrated.

-O-

Release No. 48-71